Beam Charge per Hour

Local application

Thu, Aug 4, 2005

A request was made for a device that exhibits the total Booster beam charge per hour. A continuous sum of this data is already provided as part of the millisecond sums of the Booster beam loss monitors. But referencing differences of this ever-increasing sum to provide the device requested was not deemed convenient for Acnet. An alternative is to ask the front end to do it. This note describes a new local application called `BINJ` that does this.

The new LA is designed to run in `node06C6`, the front end that includes the beam charge signal and also the 40 double precision millisecond sums of this signal. The last of the 40 values is used to hold the peak charge reading that occurs very near the time of injection from the Linac. The local application `BLMS` scans the 12.5 KHz samples around the time of injection and picks the largest value measured, assigning it as the peak value of beam charge. All of these double precision accumulations are housed in non-volatile memory; they are never reset in order to facilitate using them to measure beam loss (or beam charge) occurring between any two selected times corresponding to data-logged values.

Parameter	*Value*	*Meaning*
`ENABLE B`	`00BE`	Usual LA enable Bit#
`CHARGE C`	`0026`	Target analog channel to hold raw floating point result
`DOUBLE ADDR`	`0043` `67B8`	Memory address of double precision peak beam change sum
`SAMPCYC`	`0384`	Number of 15Hz cycles between samples of peak charge sum

The 40 millisecond beam loss (and beam charge) double precision sums are maintained for every BLM and for every Booster reset clock event. The particular one used here is that associated with clock event `0x10`, which is the "OR" of all Booster beam-accelerating clock events. It is selected by the address where it is kept. In this case, that address is `0x4367B8`, which is in the IRM non-volatile memory. The sample cycle parameter holds the number of 15 Hz cycles between successive samples of the double precision sum.

The logic used here is based upon an array of the most recent 60 samples. By specifying the sample cycle parameter above (900 decimal), the sum is sampled every minute. Keeping an array of these sums, then, covers the requested period of one hour. The result channel is updated every minute with the difference between sums sampled one hour apart.

The program waits until the next time to sample to obtain the new sum value. Before overwriting the oldest sum in the array, it subtracts the oldest sum from the new sum, which yields the beam charge that was injected during the past hour. The old value is overwritten with the new value, and the index into the circular array is advanced for use with the next sample. Although it is necessary to maintain the forever sums as doubles, the double precision difference can be stored as a single precision value. It is thus suitable for assigning to a dummy analog channel tagged to house a raw floating point value. (It is termed "raw" because there is no scaling needed; it is always maintained in engineering units.)

The end result is then updated every minute, and it always shows the total beam charge injected into the Booster over the past hour, which when first tried was about 90000 E12.

For testing, it was convenient to set the sample cycle parameter to 15, so that the samples occurred every second, and the result therefore covered the past minute.